Forensic Methods for Detecting Insider Turning Behaviors

Fred Cohen – CEO – Fred Cohen & Associates Senior member, IEEE, Livermore, CA 94550 USA

***Abstract*—This paper focuses on the use of forensic methodologies and methods for detecting subversions, an approach that may help to mitigate risk in a substantial portion of cases of types characterized to date. In essence, we look for the telltale signs of cover-ups.**

Keywords-component; insider; turning behavior; forensics; consistency analysis

I. Introduction

Malevolent or inadvertent acts by insiders may, in some cases, lead to potentially serious, grave, or catastrophic consequences. In cases where other techniques don't adequately mitigate risks and potential consequences warrant extraordinary protection, forensic approaches to detection may be among the few available options with a reasonable chance of successful risk mitigation. This paper focuses on the use of forensic methodologies and methods for detecting subversions, an approach that may help to mitigate risk in a substantial portion of cases of types characterized to date. In essence, we are looking for the telltale signs of cover-ups.

A. Background on the insider threat

When considering the "insider threat", a few basics must be defined. Insiders include anyone authorized beyond the authority of the general public. As such, there are typically many insiders, all of whom may be considered threats. We generally grant insiders authority based on trust, and in most cases historically, that trust is justified. Insiders acting commensurate with the trust placed in them by the organization granting access are "loyal". If and as insiders change loyalties while still retaining authority, we call that change in loyalty "turning".

According to statistics published by CERT, (disloyal) insiders intentionally alter and/or falsify records and/or exploit technology to avoid attribution for bad acts. Specifically, 76% of (disloyal) insiders were identified after being caught to have taken steps to conceal their identities, actions, or both, 60% compromised another user's account to carry out their acts, and 88% involved either modification or deletion of information. [1] We call these acts "subversions" because they are apparently intended to subvert normal attribution mechanisms that might otherwise lead to actors being held responsible for their acts. In many other cases not identified herein, similar patterns have been identified.

Studies over the past 15 years have also shown that particular personalities and typologies characterize insiders who betray the trust placed with them. [2][3] These include the combination of {avoidant and/or schizoid} x {anti-social and/or narcissistic and/or paranoid} individuals displaying computer dependency, entitlement, reduced loyalty, and/or ethical flexibility who experience social and/or personal frustration and lack of empathy. They undergo specific pathways before committing fraud, espionage, or sabotage, and these pathways are known. Such pathways typically include repeated subversions that increase in malice and consequences as they either go undetected, unreported, or as they are detected and reported but inadequately responded to by the administrative processes. There is a lack of sensing and fusion of the related data, and historical data seems to indicate that detection and appropriate response prior to significant harm is feasible.

A recent survey suggest that the insider threat may be quite severe, with answers to the question "Would you reveal company secrets?" producing 8% indicating "Yes, I've done it already" and 17% indicating "Yes, I have a price I'd sell for".[4] Historically, rules of thumb in the security community have been something to the effect that that 1/3 of insiders will never turn, 1/3 will turn given the belief they will not be caught, and 1/3 will actively seek opportunities to turn, and this is somewhat consistent with the identified survey.

Accurate attribution is fundamental to insider detection and proper disposition, and subversion tends to reduce the accuracy of attribution, at least for a time. But subversion behaviors also provide an opportunity to detect turning behaviors. Recent results identify that the time between the first observable indicator of an insider turning and damages with substantial consequences tends to be 6 month in 80% of cases, and is rarely less than 3 months. [5]

While rapid detection and response may be effective at detecting unauthorized and unusual acts, insiders turning typically undertake "authorized" acts and in many cases, those acts are commonplace for the identified individual. False positive rates of most anomaly and intrusion detection methodologies tends to generate "witch hunts" in which many innocent individuals are investigated and put under scrutiny, creating cultural and organizational problems and consuming scarce investigative resources. However, because there is more time between the initial observable of a subversion and the high consequence acts associated with most insider turning, time is not of the essence in the same sense as it is in real-time attack mitigation, and higher surety may be attained by a more thorough and comprehensive process. Thus the notion of using forensics approaches.

We hypothesize that, in some cases, forensic approaches may provide opportunities to intervene before bad acts are fully realized. By detecting subversion activities, investigators may gain the opportunity to develop suspicions of, observe precursors to, and limit effects of, insiders who try to defeat attribution of their acts. In particular, they may be able to stop malicious acts before they cause serious harm to the larger mission of their organization. It's not the crime, it's the cover-up.

DOI 10.1109/SPW.2012.21

B. Background on relevant forensics approaches

Methodologies for detecting acts normally involved in modifying or deleting information, using other user accounts, deleting or modifying records of acts, and actively raising suspicion of others, are largely unrelated to the tools of intrusion and anomaly detection. [6] While it is possible that an insider might use known malicious attack methods typically detected by intrusion detection methodologies and systems, doing things that trigger such systems is rarely if ever necessary for an authorized insider. Such detection techniques typically produce large numbers of false positives, and all the more so in cases where they are not tuned to known intrusions of the sorts identified. Similarly, current anomaly detection methods are poorly suited to detection of what, from the standpoint of the computer, are normal and authorized behaviors by authorized individuals. In addition, intrusion and anomaly detection systems tend to rely on data from the systems involved, and if those systems are being actively subverted, such information is not typically reliable. These issues with intrusion and anomaly detection methods and systems limit the utility of existing sensors and analysis methods to detect subversion behaviors.

Traces of activities performed by finite state automata, in the form of digital residues, are generated by the mechanisms of operating environments as an effect of the execution of those automata. [7] Because of the substantial redundancy that exists in modern operating environments, it is considered very hard to alter audit trails or other sorts of content in ways that are not detectable by examining the resulting inconsistencies between altered traces and unaltered traces of normal system behaviors. [8][9] However, there are large numbers of possible approaches to performing inconsistency analysis, [10] and specific methods for performing such analysis are limited today.

When certain sorts of inconsistencies are found, they may be used to show attribution mechanism subversion and, in some cases, they may be used to attribute those subversions to particular parties. [11][1] There are several published classes of methods for performing this sort of analysis [12][13][14] [15][16], but these general methods have limited experimental confirmation and known failure modes. The known failure modes include such problems as lack of compensation for time precision and accuracy limits, [17] non-zero base rates for apparent inconsistencies, [18] high difficulty of adequately precise characterization of the underlying automata,[19] and high computational complexity of general approaches at granularities fine enough to be applied at substantial scale.[20][21][22][23] [24]

While we cannot, in general, invert time and determine what took place from available (i.e., incomplete) traces, we can reduce the envelope of generating sequences associated with a particular set of traces by taking redundancy into account.[25][26][27[]28][29] The process of elimination can then be used to reduce the number of causal sequences and associated actors, in some cases to the point where only one individual remains as a possible suspect for a particular act. [30][31] By adding or using existing redundant traces, the set of causal chains leading to sets of traces may be substantially restricted while adding little overhead to the sensor system or operating environment.

A pleasant side effect of this approach is that, since; (1) trace inconsistency is almost always associated with a far smaller set of traces than anomaly or intrusion detection mechanisms and (2) base rates for inconsistencies tend to be very low for many such traces and processes that produce them; the volume of false positives will likely drop significantly. In many instances, the certainty of detection associated with inconsistent traces is very high (i.e., they are only reasonably attributable to intentional acts and hardware failures of particular types). In addition, such checks can be generic in nature, relying only on logical analysis based on properties of finite state automata rather than human-specified patterns or statistical analysis of human generated behaviors. A further benefit of such an approach is that it lends itself to forensic use, in that it is based on a scientific methodology that, if properly applied, can be used in a legal setting, used and introduced in court by experts, and sustained through the legal process.

Key challenges of this approach include (1) there are general theories in this area but only a few examples of algorithms that have been successfully applied, [32][33][34] [35][36][37][38] (2) performance limitations associated with the size of the overall space of such inconsistencies [4] and the known algorithms imply limited use, forcing selection of specific classes of inconsistencies to meet specific performance requirements, and (3) sensors are typically best placed close to the source of the data, but this tends to make them more susceptible to subversion, so collection of sensor data at monitoring points is likely to be a more sound architecture. This implies that joint host and infrastructure mechanisms may be required for analysis in near-real-time. Since data normalization is key to these sorts of analysis and there are existing normalization mechanisms and databases used in intrusion and anomaly detection systems, it seems likely that a solution integrating existing methods with new analytical techniques will be lowest cost and most efficient to implement in the short run.

II. Our forensics approach

Our approach involves the use of forensic analysis of redundant traces to examine data that either (1) already exists or (2) can be readily created by adding common types of sensors; to detect precursors of more damaging acts through the detection of subversions. It is based on the principal that alteration of select traces of acts in computers tends to leave inconsistent information in redundant traces. Our approach is based on a methodology in which trace inconsistency in excess of base rates is a basis for asserting that a system is not properly attributing acts to actors. With very low base rates and the fact that insiders turning or turned tend to produce these sorts of indicators with high likelihood, this offers a potential indicator of insider turning behavior, with observables well in advance of significant harm.

A. *Identification of redundant traces indicative of subversion and with low base rates*

The predominant mode of consistency analysis in attribution cases is searching for Type C inconsistencies [39] [40] (i.e., "internal" inconsistencies within records), such as missing traces indicative of periodic processes (e.g., every minute there is a log entry for a cleanup process but for a period of 4 minutes there are no such log entries) or traces in of one form (e.g., modification dates and times) that indicate different activities (e.g., last modification times) than traces of a different form. (e.g., syslog entries indicating activities that are normally logged after the last modification time) While there are various complexities in this sort of analysis (e.g., limits on precisions of times kept in different systems for different traces) [41][42][43][44], once base rates and characteristic behaviors are identified, for select traces, inconsistency detection has shown to be highly reliable. These methods have been successfully applied to email message headers, system logs, program logs, time stamps, and other similar sorts of traces.[45][46][47][48][49][50][51]

The challenge of identifying the specific sorts of traces indicative of subversions has been studied for select sorts of such acts. [52][53][54] In these cases, traces of electronic mail messages were presented as mailbox files asserted to be reflective of what took place. In earlier cases, inconsistencies had been found indicative of duplication of records, deletion of records, and alteration of message headers and bodies. [55][2] In-depth examination in previous cases revealed a number of inconsistencies that could be automatically detected, and automation was implemented to extract and analyze headers and bodies so that an examiner skilled in the art could readily detect and demonstrate these and a variety of other similar inconsistencies. [56][57] This included such methods as checking date and time stamps of records against ordering in traces, checking header information for time variations and ordering differences, and checking sequence numbers and identification headers, both of which should be monotonically increasing with time, for inconsistencies in the resulting orderings. Audit trails may also be correlated to each other and the interaction of programs with other programs correlated to the audit trails to determine if they are consistent. This approach was undertaken in the 1990s.[58] Results indicated that creating false but consistent audit trails from existing audit trails is quite difficult. In simple cases, known format for fields and records are assumed identifiable, and this is exploited to allow the analysis to be done efficiently. But complexity issues start to get more difficult as the traces are less constrained. In one case, adding and removing audit records and inconsistency in audit trails were identified, both with respect to unexpected present and missing audit records. [59] More general schemes have been proposed [60][61][62][63][64] but only limited experimental validation has been done.

These previous efforts suggest that similar methods will work for the sorts of subversions associated with recent insider attacks in which the insiders compromise another user's account to carry out their acts and/or modify or delete information. [65] For example, the deletion of log file information regarding logins does not undo the various effects of access to files like the ".profile" file in Unix environments, which will normally be changed upon login independently of any system log entries. Using the "touch" command to set the last access time on the .profile file may then add log entries associated with its execution. Altering the log files to remove this information may cause the loss of a periodic process trace that normally occurs in various log files. The list goes on and on, but as the mechanisms become less direct and more delayed, the potential for false positives and negatives presumably increases. For example, a subsequent login by the legitimate user may overwrite the file access date and time on the .profile file, making the redundant trace that might be an indicator unreliable after the next login time.

To identify redundant traces indicative of subversion, we start with historical data on insider methods, [66] [67] [68] [69] [70] [71] [72] [73] augment this with data from cases, and further augment this data with theoretical models generated through security simulation.[74] We may then use red teaming experiments, similar from an infrastructure standpoint to previous such experiments,[75] to emulate insider subversion activities and generate traces of those acts in systems similar or identical to those identified by any p[articular enterprise as the most appropriate environments for their needs. Data already exists for testing purposes for specific systems, and experiments readily generate traces associates with subversion acts by doing differentials between prior and subsequent states as were done in red teaming experiments in controlled environments.[76] This involves taking forensically sound bit-for-bit images prior to and after experiments and identifying differences in traces. These differences are then compared to differences in traces for nearly identical event sequences in which no such acts are performed to determine which traces are produced as a result of the acts of subversion. Repetition is used to increase reliability of these results, and where possible, specific mechanisms are identified as producing the traces. Some such traces may also be affected by other activities that are not necessarily present in the experiments, so once mechanisms are identified, hypothesized alternative paths to generation of similar traces are identified and experimentally tested to determine which traces are reliable indicators of the identified classes of subversion acts and which are caused by other identified non-subversion events.

This process is similar to previous forensic testing processes such as the use of cryptographic checksums on files [77] from images. Forensic testing using imaging methods where images are taken before and after activities have been in use since at least the 1990s. Reboots to identify Windows files that change over reboots were widely used to counter claims that shutting down a system corrupted content, and repeated experiments from starting images with differentials to ending images were used in deception experiments.[78] The difference is that previous efforts have not sought to track specific traces to specific mechanisms, hypothesize variations on origins that might cause them to be less reliable, and generate experiments to determine

reliability and related information that could reduce false positive and negative rates and reduce the prior state sequences that are candidates for causality.[79] While this is not expected to reduce the set of prior sequences to unity or to allow precise time reversal, previous results suggest that it will help refute challenges associated with common alternative scenarios. While digital space still generally converges with time,[80] eliminating large classes of prior event sequences remains beneficial in terms of increased certainty about prior events and makes claims regarding indications associated with specific traces more demonstrably reliable. This process produces candidates for subversion trace detection.

A specific method for detecting inconsistencies that, historically, have been used as part of cover-ups of activity, is called "JDLR", which stands for "Just don't look right". This is a method used by police from time immemorial and by cyber cops since at least the mid-1990s. It was included in a software product in 1990s[81] but was taken off the general market with the passage of the digital millennium copyright act that limited legal sales to law enforcement and government.[82] This particular method attempts to do trace typing based on header information in files and compares the typed trace to the file's extension to detect inconsistencies (e.g., a Microsoft Word® document named as a ".zip" compressed file). This sort of output is not normally produced by such software, and would appear to make it a good candidate for detection of subversion. However, in examining numerous enterprise environments, it was determined that this particular substitution has substantial non-zero base rates. Upon further investigation, it was identified that subversion of normally effective protective controls were undertaken on a regular basis by insiders in order to bypass firewall restrictions on passing various kinds of information required for business purposes. While this may not appear to be a false positive, in fact it is. An undocumented procedure in common and widely accepted use with such enterprises is the renaming of file types to specific other file types so as to bypass such protections. This is necessary in order to perform normal business functions, and lacking another acceptable method, this method is used. The prevention of files with particular extensions is widely, and ineffectively, used to limit external introduction of file types with executable content (e.g., Word® documents with macros enabled). If and to the extent that an authorized bypass mechanism that doesn't require such subversion is implemented, this technique is effective at detecting subversions of this sort.

They key element in any such approach is to identify low base rate indicators of subversions. Consider that for a network containing thousands of uses with tens of millions of files each, there are at least tens of billions of files that must be examined. Base rates of 1 in a million will produce tens of thousands of false positives, each requiring further investigation. Even at very low base rates, such detections can only reasonably be treated as presumptive positives for subversion.

B. Particularization

In most cases, ordering of real-world events are key to understanding. Normal mechanisms, such as file locks, used to force sequential output in files, may cause output from parallel processes to be entered into a file in a different order than the order in which they arrived and the time stamps were placed within those entries.[83] The inherently problematic nature of getting accurate times with similar format and precision across computers and mechanisms may also limit the precision with which ordering may be assured. [84][85][86][87][88] In time analysis, for cases where ordering variations are important, specific mechanisms at issue should be examined, and an appropriate Δ identified to limit false positives. A POset is then formed so that $\forall t_1,t_2, |t_1-t_2|<\Delta \Rightarrow t_1 \approx t_2$. This size of this POset grows exponentially with the size of Δ[89] so minimizing Δ is vital to practical use. Recent work in the analysis of overlay patterns of disk writes shows that ordering of file writes can be limited by examining existing patterns of file storage areas on disk,[90] but such analysis is quite complex and time consuming and has not been widely tested. More detailed analysis of time sequencing from traces to validate digital time-stamps has also been done,[91] but experiments showed non-zero error rates, and unexplained time deviations have been found. The key analytical issue is to gain adequate experimental evidence to bound the value of Δ.[92] While attempts to use statistical characterizations have been undertaken [93] these efforts ignore the fact that finite state automata that produce the sorts of traces of interest do not produce randomly distributed trace timings.

Early efforts [94] acknowledged but largely ignored the potential for audit trails, meta-data, and related records, to have different time bases and granularities. If one program gets time data as it starts, and another as it ends, even though they start and end together, they may produce substantially different records. Internal ordering properties must be taken into account in such analysis, but only limited studies of such consistencies have been undertaken in the published literature to date. The value for the Δ identified earlier is harder to determine if different mechanisms are involved. [95] Little progress was made in this arena between the 1990s and the 2000s, when researchers started to increasingly recognize that consistency issues were fundamental to understanding anything definitive about traces. This is largely because of the fact that digital space converges with time.[96] As a result, it becomes necessary to find redundant traces to reduce the size of the space of possible event sequences that could have produced any given set of traces to the set of FSMs that could produce all of the relevant traces in the proper sequences. This was recognized by Stallard [97] in his analysis of invariants, Carrier [98] in his attempts to run time backwards, and Gladyshev [99] in his attempts to formalize reconstruction, but not formalized as part of the physics of digital information until 2010. [100] Implementations of these methodologies have been undertaken and show promise in the sense of producing viable results in specific cases. Messaging examination is an area where these sorts of methods have born fruit. For

example, consistencies between multiple independent traces were used in attribution in [101] using methods identified in [102] and[103], and inconsistencies detected in traces of messages were indicative of fabricated duplicate claims in [104] and as discussed for other matters in [105]. Some file date and time stamps for some versions of Windows have granularity of 24 hours. [106] Studies of these issues have found many other significant differences in granularity of different audit and related records, even though the precision may be far higher than the accuracy. [107][108] Intentional subversion of time-related data is commonplace and there are widely available tools to automate file and other time stamp trace alteration, including free tools like "Timestomp". [109] Methods to detect such alterations are increasingly being developed and tested as well, again based on redundant traces. Generic methods for system-wide automated inconsistency checks have been investigated [110] but such methods are problematic for large-scale use because of the requirements to formally define all of the relevant finite state automata with proper precision and because they are inherently computationally complex.

Our approach is a bit different. Rather than trying to identify and bound time deviations for all potentially meaningful traces, we hypothesize that we can bound Δ effectively for the time values of import to reliable detection of subversion by experimenting with normal operating conditions for specific system types and selecting only trace types that have Δ sizes small enough to prevent the potentially exponential growth of the size of the resulting POset. Unlike statistical efforts such as [111], the results will be reasonably precise bounds on Δ, and unlike more generic efforts, they will produce relatively small partial orderings traceable to specific mechanisms known to be useful in detecting known classes of subversions. These results must also be experimentally validated so that reliability data will be available for potential legal use and so that analytical processes can apply the reliability information meaningfully in combination with other data for larger scale analysis, which includes reliability in its process.

For classes of time ordering inconsistencies experimentally found to be indicative of subversion, we use the same experimental methods described above to identify relevant traces that can be reasonably time bounded, use these to form bounded sized POsets, and automatically check for ordering consistencies. Ordering consistencies are be checked by identifying specific invariants with respect to the traces identified rather than by the creation of generic invariants, but otherwise use methods similar to those of Gladyshev [112] and Stallard [113]. Finite automata models are created for identified reliable traces and traces that appear will be extracted in sequence with time-related information and run against the FSM models to detect state transitions not appearing in the models. These are identified as inconsistent with normal behaviors. To the extent that specific sequences are associated with specific known subversions, finite automata for those subversions are created and sequences identified as being inconsistent with normal behaviors run against known subversion behaviors to produce more definitive positive indicators and particularize them to specific methods.

We have taken the tact of performing experiments on specific inconsistencies detected as presumptive positives. In particular, when identifying a presumptive positive inconsistency (e.g., a ".doc" file hidden as a ".zip" file) with a particular set of date and time stamps associated with file creation, access, and modification, we perform testing to determine which of the known methods of changing a ".doc" file to a ".txt" file are consistent with the particular time attributes. Testing can be fully automated so that, for example, if known methods of making such a change are the "copy" command, the "rename" command, and the use of a graphical interface, each can be experimentally tested programmatically with resulting timestamps determined and compared to actual attributes found to particularize the possible methods by which the subversion may have been done. Programs like "Timestomp" and other similar mechanisms also have side effects that may be tested and sought to associate particular traces with their use.

C. Individualization

Yet more and more revealing results may be generated by examining for both type C and type D (i.e., "external" inconsistencies between records) consistency. [114][115] When sworn statements are found to be inconsistent with traces, problems arise with the credibility the traces and/or the witness. If there is also type C inconsistency, the witness may use this as a basis for claiming that the traces are invalid. In the case of an investigative process, this is typically a complex situation involving a lot of human time and effort, and definitive answers are hard to find. That's why juries are used to make such judgments. Such approaches are usually not amenable to automation and are not highly scalable. But there are external event sequences that may be automatically analyzed for type D consistency. For example, external records of presence (e.g., badge entry and exit records) may be matched to internal records of use (e.g., audit records of activities by the individual's user identity) and, in cases where there is an inconsistency (e.g., the badge records indicate that the individual was in the lunch room from 12:30 to 13:00 while the audit record indicates use of their account from another area at 12:45), indications of insider subversion are indicated. By using additional external traces (e.g., the records of all individuals in each area over time) and internal traces (e.g., others who were logged in or performing other activities in the relevant time frames) the process of elimination may be used to produce more definitive attribution (e.g., the actor who was present and no alibi on all of the 8 detected subversion instances is a far better suspect).

The computer security field started to work to integrate sensors associated with badging and other physical systems in the late 1990s and early 2000s. Such companies as Netbots offered IP-based systems with built-in sensors that combined video, sound, temperature, CO_2, motion, infrared, smoke, and other similar devices in a small hardware platform that could be integrated with emergency, alarm, and entry systems, as well as identity management associated

with computer-based access control mechanisms. Large companies like Computer Associates promised the ability to provide fully integrated controls for intrusion and anomaly detection, but this has not been widely realized in an integrated and fully automated system. Rather, alarm, anomaly, and intrusion systems are typically fused for use in security operations centers where people participate in real-time analysis of and response to detections. In investigative processes, these sorts of information are commonly combined, and the combined traces are used as part of the examination process to identify consistencies and inconsistencies. But scaling these sorts of activities has not been substantially pursued in the literature and attempts to bring these sorts of mechanisms to the commercial markets have largely failed, in part at least because of (1) the need to customize analysis and response to the particular environment, and (2) the lack of trust for automation to make decisions with high consequences uninhibited by executive management decisions.

Our approach avoids many of the pitfalls of large-scale broad spectrum integration of type C and D trace consistency analysis by (A) focusing in on specific indicators identified based on insider subversion methods already identified, [116][117][118][119][120] (B) more specifically limiting our focus to such indicators as are related to identified subversion methods with low base rates and small Δt values, and (C) only applying such indicators as already exist in digital form. There may be few such indicators in any particular environment, which reduces the complexity of the analytical process, and we suspect that their use will further tie down the presence of attempts at subversion and help to more definitively identify the likely insider violating trust.

Historical reviews of records from previous investigations indicate that individualization is often feasible in such cases, typically through the process of elimination in the use of multiple data sources. While all of this data is rarely available in near-real-time today, it appears that improvements in automation makes automated individualization using these methods feasible within the time frame of weeks to months.

D. The addition of select redundant sensors and traces to enhance detection

In order to detect inconsistencies and reliably attribute acts to actors, it may also be helpful to add redundant sensors that more readily reveal inconsistencies associated with attempts at subversion and to increase the certainty of attribution of subversion attempts to their sources.

The attribution of actions to actors when it comes to subversion is often characterizeable by the limits on actors who have adequate control to perform the subversion. A method for forensic analysis of control [121] has been applied to limited legal matters and holds promise for further application in this arena. This method is based on the principal that in order for an actor to intentionally control a system or mechanism they must have (1) the ability to act so as to express intent, and (2) the ability to have that expressed intent carried out. If we apply this to subversion, it is clear that anyone who carries out an intentional subversion must have these two things.

In order to perform the analysis, this is broken down into cases where there is (0) no control, (1.1) direct control, and (1.2) indirect control. No control is demonstrated by a lack of either syntax to express identified intent (i.e., the act is thus outside of the syntactic control envelope) or authority to carry out intent (i.e., the act is thus outside the semantic control envelope of the actor). Direct control (i.e., evidence supporting violation) is either through a special or general (i.e., finite Turing equivalence) purpose mechanism in normal or abnormal (i.e., the normal control envelope of the mechanism is exceeded by either an exploited weakness or an uncovered path) use. [122][123] In any case, traces must evidence the use of syntax to express the intent and the semantic effect of the expressed intent in order to show control. Indirect control is demonstrated by identifying a mechanism, by which a new control envelope may be entered (e.g., by gaining general purpose access to the enveloping machine from the inside of a virtual machine). While this analysis is currently a purely human activity in the general case, for specific cases where a methodology is presupposed for carrying out the controlling acts (e.g., a systems administrator uses their privileged access to delete traces from an audit trail) direct control may be easily shown feasible (i.e., it is feasible for a systems administrator to do this with their normal systems administration access). Thus the syntax is available for expressing the intent and if expressed the intent will be carried out. The problem that remains is demonstrating that such syntax was in fact presented and that such intent was in fact carried out.

We use a different approach than is used for the general case. While inconsistency may be apparent as a result of the mechanisms and analysis described earlier, attribution is more complex. Rather than trying to prove each attribution individually, we create special purpose sensors specifically designed to detect the acts of subversion associated with insiders. These sensors are specifically oriented toward attribution of subversion behaviors to responsible actors. For example, a sensor added to codify elements of the process lineage [124] and cryptographic checksums of the last several minutes of traces in system log files at pseudo-random intervals less than a minute apart provides redundant indicators of altered log files and processes associated with subversion behaviors. Deletion or alteration of these log entries is readily detectable by their absence at the known (pseudo-random) time and the lack of subsequent trace consistency with later cryptographic checksums covering the prior entries. Such sensors have to meet the same requirements as other similar traces used as indicators (i.e., they have to be qualifying traces with time variations within tolerance to limit POset sizes), each of which may be verified by the same means as the previous methods discussed. By adding new or better applying existing redundant traces, attempts to subvert attribution may be more easily found and, in many cases, proper attribution of the subversion made. Complexity of analysis and time to detect may also be substantially reduced by adding select redundancy specifically designed for this purpose. When the

time of human acts can be constrained this closely and detected this quickly, it is also often feasible to use type D consistency checks to individualize the actor for attribution.

III. Summary, conclusions, and further work

Based on historical data and prior studies, we have discussed a method by which insiders may be detected in adequate time to mitigate the more dire consequences typically realized. This method uses a forensic approach to detecting low base-rate Type C inconsistencies producing presumptive positives. Automated testing against hypothesized causes produces limited particularization to causes consistent with available traces, subject to the infeasibility of general time reversal. Individualization is produced by the use of Type D consistency checks against external redundant records and the process of elimination. All of this is done while taking account of the limited precision and accuracy of time as recorded in digital systems. Added traces are identified as a method to reduce the difficulty and time to detection, and to increase the surety of detection, particularization, and individualization.

It appears that this method may be effective at automating forensic approached to higher surety detection of insider subversions associated with turning behaviors and do so in time to mitigate the most serious negative consequences of those acts.

References

[1] M. Keeney, E. Kowalski, D. Cappelli, A. Moore, T, Shimeall, S. Rodgers, "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors", Jan 2005.

[2] E. Shaw, K. Ruby, and J. Post, "Insider Threats to Critical Information Systems: Typology of Perpetrators, Security Vulnerabilities, Recommendations", Aug 31, 1999. ASD-C3I – OIO - Contract # 98-G-7900, Task Letter Number 001:Insider Threat Profile.

[3] E. Shaw, K. Ruby, and J. Post, "Insider Threats to Critical Information Systems: Characteristics of the Vulnerable Critical Information Technology Insider (CITI)" Contract Nr. N39988-97C-7850, Sep 25, 1998.

[4] J. Yang and K. Gelles, "Poll reveals 75% prefer an honest day's work" in USA Today, November 18, 2011, citing a Monster.com survey.

[5] Personal communicaiton with researchers on a pre-publication result.

[6] F. Cohen, "Intrusion Detection and Response Systems", Burton Group Report, October, 2003

[7] I.b.i.d. "Digital Forensic Evidence Examination", Chapter 3

[8] F. Cohen, "A Note on Detecting Tampering with Audit Trails", 1995, Available at URL: http://all.net/books/tech/audmod.pdf

[9] F. Cohen, "Analysis of redundant traces for consistency", IEEE International Workshop on Computer Forensics in Software Engineering (CFSE 09), Seattle, Washington, USA, July 20-24, 2009.

[10] F. Cohen, "Two models of digital forensic examination", IEEE SADFE (in conjunction with the IEEE Oakland Conference), 2009-05-21, Oakland, CA

[11] F. Cohen, "A Method for Forensic Analysis of Control", IFIP TC11, Computers & Security, V29#8, pp 891-902, Nov., 2010, doi: 10.1016/j.cose.2010.05.003

[12] B. Carrier, "A Hypothesis-Based Approach to Digital Forensic Investigations", Dissertation, Purdue, CERIAS Tech Report 2006-06, 2006.

[13] P. Gladyshev, "Formalising Event Reconstruction in Digital Investigations", Dissertation, University College Dublin, 2008.

[14] F. Cohen, "A Case Study in Forensic Analysis of Control", Journal of Digital Forensics, Security, and the Law, 2011.

[15] S. Willassen, Finding Evidence of Antedating in Digital Investigations", ARES 2008 The Third International Conference on Availability, Reliability and Security, 2008.

[16] P. Gladyshev and A. Enbacka, "Rigorous Development of Automated Inconsistency Checks for Digital Evidence Using the B Method ", International Journal of Digital Evidence, Fall 2007, Volume 6, Issue 2.

[17] F. Cohen, "Digital Forensic Evidence Examination", [Note: Chap 3 -" The physics of digital information" is available online at http://infophys.com/InfoPhys.pdf] ASP Press, 2009-2011.

[18] Svein Yngvar Willassen, "Timestamp Evidence Correlation", Presentation at IFIP WG 11.9 International Conference on Digital Forensics, January, 2008.

[19] I.b.i.d. "Digital Forensic Evidence Examination"

[20] F. Cohen, "Analysis of redundant traces for consistency", IEEE International Workshop on Computer Forensics in Software Engineering (CFSE 09), Seattle, Washington, USA, July 20-24, 2009.

[21] B. Carrier, "A Hypothesis-Based Approach to Digital Forensic Investigations", Dissertation, Purdue, CERIAS Tech Report 2006-06, 2006.

[22] P. Gladyshev, "Formalising Event Reconstruction in Digital Investigations", Dissertation, University College Dublin, 2008.

[23] F. Cohen, "A Note on Detecting Tampering with Audit Trails", 1995, Available at the URL: http://all.net/books/tech/audmod.pdf

[24] Svein Yngvar Willassen, "Hypothesis-based investigation of digital timestamps", chapter in Advances in Digital Forensics IV, Ray and Shenoi ed., Springer, ISBN# 978-0-387-84926-3, 2008.

[25] S. Willassen, Finding Evidence of Antedating in Digital Investigations", ARES 2008 The Third International Conference on Availability, Reliability and Security, 2008.

[26] T. Stallard, "Automated Analysis for Digital Forensic Science", Masters Thesis, Computer Science Department, University of California Davis, 2002.

[27] I.b.i.d., "Analysis of redundant traces for consistency"

[28] I.b.i.d., "A Method for Forensic Analysis of Control"

[29] I.b.i.d., "Formalising Event Reconstruction in Digital Investigations"

[30] F. Cohen, "A Method for Forensic Analysis of Control", IFIP TC11, Computers & Security, V29#8, pp 891-902, Nov., 2010, doi: 10.1016/j.cose.2010.05.003

[31] "Defendant's expert witness disclosure of Dr. Frederick B. Cohen", Susan Polgar v. United States of America Chess Federation et. al., C.A. NO. 5-08CV0169-C in the United States District Court – Northern District of Texas, Lubbock Division. 2009-09-15

[32] I.b.i.d., "A Method for Forensic Analysis of Control"

[33] I.b.i.d., "A Hypothesis-Based Approach to Digital Forensic Investigations"

[34] I.b.i.d., "Formalising Event Reconstruction in Digital Investigations"

[35] I.b.i.d., "A Case Study in Forensic Analysis of Control"

[36] S. Willassen, Finding Evidence of Antedating in Digital Investigations", ARES 2008 The Third International Conference on Availability, Reliability and Security, 2008.

[37] I.b.i.d., "Rigorous Development of Automated Inconsistency Checks for Digital Evidence Using the B Method"

[38] I.b.i.d., "Automated Analysis for Digital Forensic Science"

[39] I.b.i.d., "Analysis of redundant traces for consistency"

[40] F. Cohen, "Two models of digital forensic examination", IEEE SADFE (in conjunction with the IEEE Oakland Conference), 2009-05-21, Oakland, CA

[41] I.b.i.d., "Error Rates for Timestamp Forensic Analysis"

[42] I.b.i.d., "Digital Forensic Evidence Examination"

[43] http://msdn.microsoft.com/en-us/library/ms724284.aspx "Not all file systems can record creation and last access time and not all file systems record them in the same manner. For example, on NT FAT, create time has a resolution of 10 milliseconds, write time has a

resolution of 2 seconds, and access time has a resolution of 1 day (really, the access date). On NTFS, access time has a resolution of 1 hour."
[44] C. Boyd and P. Forster, "Time and date issues in forensic computing--a case study", Digital Investigation (2004) pp. 18-23, 2004.
[45] I.b.i.d., "A Case Study in Forensic Analysis of Control"
[46] I.b.i.d., "A Note on Detecting Tampering with Audit Trails"
[47] I.b.i.d., "defendant's expert witness disclosure of Dr. Frederick B. Cohen"
[48] F. Cohen, "Identifying and Attributing Similar Traces with Greatest Common Factor Analysis", (Submitted 2010)
[49] William Silverstein v. Liquid Minds, LLC, et. al. Case No. BC375173 in the United States District Court for the District of Maryland. 2010.
[50] C. Boyd and P. Forster, "Time and date issues in forensic computing--a case study", Digital Investigation (2004) pp. 18-23, 2004.
[51] I.b.i.d., "Automated Analysis for Digital Forensic Science"
[52] "Defendant's expert witness disclosure of Dr. Frederick B. Cohen", Susan Polgar v. United States of America Chess Federation et. al., C.A. NO. 5-08CV0169-C in the United States District Court – Northern District of Texas, Lubbock Division. 2009-09-15.
[53] I.b.i.d., "Identifying and Attributing Similar Traces with Greatest Common Factor Analysis"
[54] I.b.i.d., William Silverstein v. Liquid Minds, LLC, et. al.
[55] F. Cohen, "Issues and a case study in bulk email forensics", Fifth annual IFIP WG 11.9 International Conference on Digital Forensics, 2009/01/27, published as "Bulk Email Forensics" in the conference publication.
[56] I.b.i.d., "Identifying and Attributing Similar Traces with Greatest Common Factor Analysis"
[57] I.b.i.d., "Issues and a case study in bulk email forensics"
[58] I.b.i.d., "A Note on Detecting Tampering with Audit Trails"
[59] .b.i.d., "A Note on Detecting Tampering with Audit Trails"
[60] I.b.i.d., "A Hypothesis-Based Approach to Digital Forensic Investigations"
[61] I.b.i.d., "Formalising Event Reconstruction in Digital Investigations"
[62] P. Gladyshev and A. Enbacka, "Rigorous Development of Automated Inconsistency Checks for Digital Evidence Using the B Method ", International Journal of Digital Evidence, Fall 2007, Volume 6, Issue 2.
[63] Svein Yngvar Willassen, "Hypothesis-based investigation of digital timestamps", chapter in Advances in Digital Forensics IV, Ray and Shenoi ed., Springer, ISBN# 978-0-387-84926-3, 2008.
[64] I.b.i.d., "Automated Analysis for Digital Forensic Science"
[65] I.b.i.d., "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors"
[66] I.b.i.d., "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors"
[67] I.b.i.d., "Insider Threats to Critical Information Systems: Typology of Perpetrators, Security Vulnerabilities, Recommendations"
[68] I.b.i.d., "Insider Threats to Critical Information Systems: Characteristics of the Vulnerable Critical Information Technology Insider (CITI)"
[69] J. Post, "The Anatomy of Treason", Psychological contributions to selective targeting, pp 35-37.
[70] E. Shaw, K. Ruby, and J. Post, "The Insider Threat to Information Systems - The Psychology of the Dangerous Insider ", Security Awareness Bulletin, No. 2-98, 1998.
[71] J. Post, personal communications - extracted from a contemporary undisclosed article.
[72] F. Cohen, "Frauds, Spies, and Lies, and How to Defeat Them", ASP Press, 2005.
[73] F. Cohen, "Defending Against the Evil Insider", Burton Group Report, November 2005.
[74] F. Cohen, "Simulating Cyber Attacks, Defenses, and Consequences", IFIP-TC11, `Computers and Security', 1999, vol. 18, no. 6, pp. 479-518(40).
[75] F. Cohen, I. Marin, J. Sappington, C. Stewart, and E. Thomas "Red Teaming Experiments with Deception Technologies", 2001. http://all.net/journal/deception/RedTeamingExperiments.pdf
[76] I.b.i.d., "Red Teaming Experiments with Deception Technologies"
[77] F. Cohen, "A Cryptographic Checksum for Integrity Protection", IFIP-TC11 "Computers and Security", V6#6 (Dec. 1987), pp 505-810.
[78] I.b.i.d., "Red Teaming Experiments with Deception Technologies"
[79] I.b.i.d., "A Case Study in Forensic Analysis of Control"
[80] I.b.i.d., "Digital Forensic Evidence Examination"
[81] F. Cohen, 1997 – ForensiX – Digital Forensics ToolKit for Linux and Unix
[82] "The Digital Millenium Copyright Act" http://www.copyright.gov/legislation/dmca.pdf
[83] I.b.i.d., "Digital Forensic Evidence Examination"
[84] I.b.i.d., "A Note on Detecting Tampering with Audit Trails"
[85] I.b.i.d. http://msdn.microsoft.com/en-us/library/ms724284.aspx
[86] C. Boyd and P. Forster, "Time and date issues in forensic computing--a case study", Digital Investigation (2004) pp. 18-23, 2004.
[87] M. Stevens, "Unification of relative time frames for digital forensics", Digital Investigation, Volume 1, Issue 3, 2003.
[88] I.b.i.d., "Automated Analysis for Digital Forensic Science"
[89] I.b.i.d., "Digital Forensic Evidence Examination"
[90] Svein Yngvar Willassen, "Timestamp Evidence Correlation", Presentation at IFIP WG 11.9 International Conference on Digital Forensics, January, 2008.
[91] I.b.i.d., "Hypothesis-based investigation of digital timestamps"
[92] I.b.i.d., "Digital Forensic Evidence Examination"
[93] I.b.i.d., "Error Rates for Timestamp Forensic Analysis"
[94] I.b.i.d., "A Note on Detecting Tampering with Audit Trails"
[95] I.b.i.d., "Digital Forensic Evidence Examination"
[96] I.b.i.d., "Digital Forensic Evidence Examination"
[97] I.b.i.d., "Automated Analysis for Digital Forensic Science"
[98] I.b.i.d., "A Hypothesis-Based Approach to Digital Forensic Investigations"
[99] I.b.i.d., "Formalising Event Reconstruction in Digital Investigations"
[100] I.b.i.d., "Digital Forensic Evidence Examination"
[101] I.b.i.d., "Defendant's expert witness disclosure of Dr. Frederick B. Cohen"
[102] I.b.i.d., "Identifying and Attributing Similar Traces with Greatest Common Factor Analysis"
[103] I.b.i.d., "Issues and a case study in bulk email forensics"
[104] I.b.i.d., William Silverstein v. Liquid Minds, LLC, et. al.
[105] I.b.i.d., "Issues and a case study in bulk email forensics"
[106] I.b.i.d., http://msdn.microsoft.com/en-us/library/ms724284.aspx
[107] C. Boyd and P. Forster, "Time and date issues in forensic computing--a case study", Digital Investigation (2004) pp. 18-23, 2004.
[108] M. Stevens, "Unification of relative time frames for digital forensics", Digital Investigation, Volume 1, Issue 3, 2003.
[109] Timestomp is a utility co-authored by developers James C. Foster and Vincent Liu. The software's goal is to allow for the deletion or modification of time stamp-related information on files., ... http://www.forensicswiki.org/wiki/Timestomp, 2005.
[110] P. Gladyshev and A. Enbacka, "Rigorous Development of Automated Inconsistency Checks for Digital Evidence Using the B Method ", International Journal of Digital Evidence, Fall 2007, Volume 6, Issue 2.
[111] I.b.i.d., "Error Rates for Timestamp Forensic Analysis"
[112] I.b.i.d., "Rigorous Development of Automated Inconsistency Checks for Digital Evidence Using the B Method"
[113] I.b.i.d., "Automated Analysis for Digital Forensic Science"
[114] I.b.i.d., "Analysis of redundant traces for consistency"
[115] I.b.i.d., "Two models of digital forensic examination"
[116] I.b.i.d., "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors"

[117] I.b.i.d., “Insider Threats to Critical Information Systems: Typology of Perpetrators, Security Vulnerabilities, Recommendations”
[118] I.b.i.d., “Insider Threats to Critical Information Systems: Characteristics of the Vulnerable Critical Information Technology Insider (CITI)”
[119] I.b.i.d, “The Insider Threat to Information Systems - The Psychology of the Dangerous Insider”
[120] I.b.i.d. personal communications - extracted from a contemporary undisclosed article.
[121] I.b.i.d., "A Method for Forensic Analysis of Control"
[122] I.b.i.d., "A Method for Forensic Analysis of Control"
[123] I.b.i.d., "A Case Study in Forensic Analysis of Control"
[124] F. Cohen, “Method and Apparatus for Providing Deception and/or Altered Execution of Logic in an Information System US Pat. 7,296,274.